2010 Third International Conference on Information and Computing

Modeling and Simulation for Material Supply in Emergent Disaster Based on Agent-DEVS/HLA

Qi CAO
Department of Training
Logistical Engineering University
401311 Chongqing, China
roy1976@163.com

Zhong-shi HE Lei YU
College of Computer Science
Chongqing University
400030 Chongqing, China
zshe@cqu.edu.cn, yl991377@yahoo.com.cn

***Abstract*—At present, the theory and technique of modeling and simulation mainly carry out special applications in emergent disaster. They are lacking in expansibility and inconsiderate to reusability and interoperability. So they can not satisfy the simulation training needs of material supply in emergent disaster. Firstly, the simulation entities of material supply in emergent disaster were analyzed. The simulation models were established based on Agent-DEVS. And the distributed interactive simulation was implemented based on HLA. Finally, the simulation test was put forward. The result shows that Agent-DEVS models may suitably describe intelligent and cooperative actions of material supply in emergent disaster. And HLA can enhance the reusability and interoperability of Agent-DEVS models. It lays the foundation for the development of simulation training of material supply in emergent disaster.**

***Keywords*—simulation modeling; agents; discrete event system specification; high level architecture; material supply**

I. INTRODUCTION

In emergent disaster, it is hard to quantitatively analyze and describe the actions among members because of unpredictability and unrepeatability. And the environment elements are complicated and changeable. So it is difficult to carry out actual rehearsing and training. Inevitably, simulation training founded on material supply models will become effective means to foster emergent succor. At present, the researches on theory and technique of modeling and simulation applied to emergent disaster are few and almost overseas. For example, Balasubramanian et al presented a simulation framework for emergency response drills — Drillsim [1]. Mondlane formulated an integrated model for the Mozambique floods to create added value for the national response mechanisms in case of emergency [2]. Balbis et al put forward a decisional model for dynamic allocation of resources in natural disasters management [3]. Each study had its strong point and special application field. But they are lacking in expansibility and inconsiderate to reusability and interoperability. So they can not satisfy the simulation training needs of material supply in emergent disaster. Interior researches emphasized particularly on analysis of simulation training, such as mine safety, landslide disaster and fire fighting [4-6]. None but we ever introduced classical DEVS into this field and put forward simulation test models in CD++ [7].

As a result, we will establish simulation models for material supply in emergent disaster based on Agent-DEVS which is an extended DEVS formalism, we proposed [8], for intelligent modeling and simulation. And then the distributed interactive simulation for these models will be implemented based on HLA.

II. ANALYSIS OF SIMULATION ENTITIES

The main simulation entities for material supply in emergent disaster include *headquarters*, *servicer*, *cooperator*, *supplier*, *supportobj* and *environment* [9].

Headquarters is the command unit in the whole succor activity. It is responsible for receiving and informing the disaster situations, harmonizing and organizing the service resources, giving the supply commands and collecting the reports from different support entities.

Servicer is the central goal of simulation. It is responsible for receiving the supply tasks from *headquarters*, planning and computing the amount of supply tasks, organizing *supplier* with the materials, corresponding with *cooperator* for the material transportation, inspection and management via *headquarter*.

Cooperator consists of traffic, sanitation, storage supply entities and so on. It assists *servicer* to accomplish the material transportation, inspection and management.

Supplier consists of warehouse, factory, market and so on. It assists *servicer* to prepare the materials and equipments.

Supportobj is the receiver of material supply. It generates and modifies the material requirement according to the development of situation.

Environment includes transaction environment and natural environment. The former comprises information generated by the emergent transaction, and the latter comprises geographical and meteorologic information.

III. DESIGN OF SIMULATION MODELS

A. Simulation Flow and Coupled Model

Above-mentioned entities can be modeled into different Agent-DEVS atomic models. Based on them, the Agent-DEVS coupled model will be created, as shown in Fig. 1.

978-0-7695-4047-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ICIC.2010.339

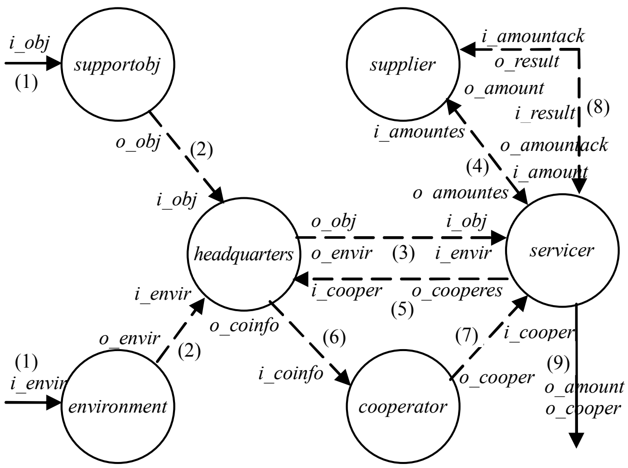


Figure 1. Coupled model of material supply in emergent disaster

Step 1: Messages with variables of support object and environment are sent to the input ports of coupled model, and then got by *supportobj* and *environment*, respectively.

Step 2: *Supportobj* and *environment* submit the supply requirement to *headquarters*.

Step 3: According to the situation, *headquarters* gives the supply command to *service*.

Step 4: *Servicer* plans the amount of supply task according to input message and prepares materials from *supplier*. *Supplier* produces the amount of planned supply based on its own state and returns to *servicer*.

Step 5: *Servicer* drafts the preparatory supply scheme and reports the amount of cooperative task for the other models to *headquarters*.

Step 6: *Headquarters* gives the cooperative command to relative *cooperator*.

Step 7: *Cooperator* produces cooperative supply scheme based on its own state and returns to *servicer*.

Step 8: *Servicer* confirms the actual amount of material supply according to cooperative message and sends it to *supplier* for acknowledgement. *Supplier* hereby modifies its own state and returns to *servicer*.

Step 9: *Servicer* sends the final amounts of supply task and cooperative task to the output ports of coupled model. End the simulation.

B. Demonstration of Atomic Models

Servicer is given as an example of all atomic models. The model is shown as follows:

$$servicer = <X, Y, P, A, \delta_{int}, \delta_{ext}, \delta_{con}, \lambda, ta> \qquad (1)$$

X = {*ip*∈{*i_obj*, *i_envir*, *i_amount*, *i_cooper*, *i_result*}, *im*∈*Msg*};

Y = {*op*∈{*o_amountes*, *o_cooperes*, *o_amountack*, *o_amount*, *o_cooper*}, *om*∈*Msg*};

P = {*agtid*∈*N*, *agtname*∈*String*, *agtlocation*∈*Position*, *agtstate*∈{*awaiting* //awaiting supply task, *computing* //computing planned amount of supply task, *computpost* //preparing materials from *supplier*, *planning* //computing planned amount of cooperative task, *planpost* //asking *headquarters* for cooperation, *implement* //confirming actual amount of material supply, *implepost* //acknowledging amount with *supplier*, *finishing* //exporting final supply result}, *agtfeasiblev*∈{*vames* //planned amount of supply task, *vcoes* //planned amount of cooperative task, *vam* //actual amount of material supply, *vco* //actual amount of cooperative task, …}};

A = {*apperception*∈$X \cup P$.*agtfeasiblev*, *disposal*∈{*compute_vames* //generating function for planned amount of supply task, *plan_vcoes* //generating function for planned amount of cooperative task, *compute_vam* //generating function for actual amount of supply task, *plan_vco* //generating function for actual amount of cooperative task, *operate_knowledge* //operating function of knowledge repository}, *operation*∈{*p_trans* //internal personality transition linked with δ_{ext}, *o_response* //output response linked with *P.agtfeasiblev* }, *knowledge*∈{*supportobj_partab* //parameter table of support object, *envirimpact_partab* //effect parameter table of environment, *materitech_partab* //technique parameter table of supply material, *cooperform_partab* //performance parameter table of cooperative material, *schedule_partab* //parameter table of schedule plan, *nearmodel_partab* //parameter table of near models, …}};

δ_{int} = {δ_{int}(*p*(…, *computing*,*vames*)) = *p*(…, *computpost*, ϕ), δ_{int}(*p*(…, *planning*,*vcoes*)) = *p*(…, *planpost*, ϕ), δ_{int}(*p*(…, *implement*,*vam*)) = *p*(…, *implepost*, ϕ), δ_{int}(*p*(…, *finishing*, *vam*+*vco*)) = *p*(…, *awaiting*, ϕ)};

δ_{ext} = {δ_{ext}(*p*(…, *awaiting*, ϕ), *e*, *i_obj*+*i_envir*) = *p*(…, *computing*, *vames*), δ_{ext}(*p*(…, *computpost*, ϕ), *e*, *i_amount*) = *p*(…, *planning*, *vcoes*), δ_{ext}(*p*(…, *planpost*, ϕ), *e*, *i_cooper*) = *p*(…, *implement*, *vam*), δ_{ext}(*p*(…, *implepost*, ϕ), *e*, *i_result*) = *p*(…,*finishing*, *vam*+*vco*)};

δ_{con} = {δ_{con}(*p*(…, *computing*, *vames*), *i_obj*+*i_envir*) = *p*(…, *computpost*, ϕ), δ_{con}(*p*(…, *planning*, *vcoes*), *i_amount*) = *p*(…, *planpost*, ϕ), δ_{con}(*p*(…, *implement*, *vam*), *i_cooper*) = *p*(…, *implepost*, ϕ), δ_{con}(*p*(…, *finishing*, *vam*+*vco*), *i_result*) = *p*(…, *awaiting*, ϕ)};

λ = {λ(*p*(…, *computing*, *vames*)) = (*o_amountes*, *vames*), λ(*p*(…, *planning*, *vcoes*)) = (*o_cooperes*, *vcoes*), λ(*p*(…, *implement*, *vam*)) = (*o_amountack*, *vam*), λ(*p*(…, *finishing*, *vam*+*vco*)) = (*o_amount*, *vam*)+(*o_cooper*, *vco*)};

ta = {*ta*(*p*(…, *awaiting*, ϕ)) = *ta*(*p*(…, *computpost*, ϕ)) = *ta*(*p*(…, *planpost*, ϕ)) = *ta*(*p*(…, *implepost*, ϕ)) = $+\infty$, *ta*(*p*(…, *computing*, *vames*)) = *COMPUTING -TIME*, *ta*(*p*(…, *planning*, *vcoes*)) = *PLANNING -TIME*, *ta*(*p*(…, *implement*, *vam*)) = *IMPLEMENT -TIME*, *ta*(*p*(…, *finishing*, *vam*+*vco*)) = *FINISHING –TIME*}.

IV. DESIGN OF DISTRIBUTED INTERACTION

A. Federation Structure

The distributed interactive structure of federation is grounded on HLA/RTI, which is shown as Fig. 2. Each federate is Agent-DEVS model and the ports of model are converted into data objects of HLA. The structure definition in Agent-DEVS, such as knowledge repository, is regarded as the object class declaration in HLA. So the mapping between knowledge update in Agent-DEVS and attribute update in HLA is established. The coupling relationship in Agent-DEVS is regarded as the interaction class declaration in HLA. So the mapping between models coupling in Agent-DEVS and instances interaction in HLA is established too. Each federate not only accomplishes its own simulation task but also interacts with other federates by RTI, which represent the physical data stream in HLA. At the same time, the couplings among Agent-DEVS models represent the potential logic data stream.

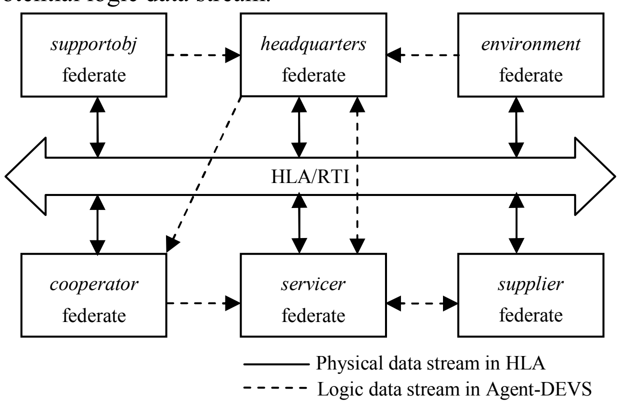


Figure 2. Structure of federation for material supply in emergent disaster

B. Design of Object Class and Interaction Class

The design of federates is different from that of typical HLA/RTI application. The description information of Agent-DEVS models should be converted into object information of OMDT. The design demonstrations of object class and interaction class are shown as TAB. I and TAB. II.

TABLE I. DESIGN DEMONSTRATION OF OBJECT CLASS

Class	Attribute	SuperClass	Class	Attribute
Materiobj	*ID*		*Envirimpact*	*Temperature*
	Type			*Humidity*
	Quantity			*X,Y(Coordinate)*
	Quality	*Materiobj*	*Materitech*	*Capacity*
Supportobj	*Age*			*Volume*
	Amount	*Materiobj*	*Cooperform*	*Freight*
	State			*Speed*

TABLE II. DESIGN DEMONSTRATION OF INTERACTION CLASS

Class	Parameter	Class	Parameter
Assign_task	*Tasktype*	*Plan_material*	*Materitype*
	Objtype		*Objamount*
	Enviritype		*Materiamount*
	Tasklevel	*Askfor_cooperation*	*Coopertype*
	Tasktime		*Cooperamount*

V. SIMULATION TEST

A simulation test based on simplified process of material supply in emergent disaster is carried out. The main hypotheses are shown as follows:

Hypothesis 1: *Tasktype* is the tents supply. *Objtype* is decided by *amount* in *supportobj*. *Enviritype* is decided by climate which lies with the *temperature* and *humidity* in *envirimpact*. *Materitype* is tents and *materiamount* is related with tent capacity, i.e., *varbase*, which is stored in *materitech_partab* of knowledge repository and corresponds to *capacity* in *materitech*. The variable of supply task, i.e., *vames*, is computed by generating function *compute_vames*, which is shown as follows:

$$vames = compute\_vames(i\_obj, i\_envir) = \text{Divide}(\text{Multiply}(i\_obj, i\_envir), varbase) \quad (2)$$

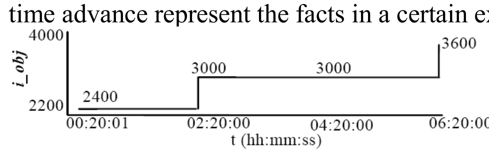
Hypothesis 2: The transportation of trucks is uniquely considered as cooperative supply. *Cooperamount* is related with transportation capability, i.e., *vartrans*, which is stored in *cooperform_partab* of knowledge repository and corresponds to *freight* in *cooperform*. The variable of cooperative task, i.e., *vcoes*, is computed by generating function *plan_vcoes*, which is shown as follows:

$$vcoes = plan\_vcoes = \text{Divide}(vames, vartrans) \quad (3)$$

Hypothesis 3: *Supplier* is a material supply station, in which the initial amount of tents in stock, i.e., *varstor*, is 1000. *Cooperator* is a truck transportation team, in which the initial amount of available trucks, i.e., *varsup*, is 150. Both of them correspond to *quantity* in *materiobj*.

Hypothesis 4: The inactive state in time advance is represented in elapsed time 99:59:59:999. And the active state is represented in elapsed time 00:10:00:000, which is set according to the desired dealing time in emergent disaster.

The running status of *servicer* federate was chiefly observed, as shown in Fig. 3. *Headquarters* federate assigned tasks to *servicer* federate in four discrete time points (Fig. a-b). The planned amounts of supply task and cooperative task were directly computed according to its own knowledge (Fig. c-f). After the negotiation and confirmation with *cooperator* federate and *supplier* federate, *servicer* federate output the actual amounts of supply task and cooperative task (Fig. g-j). Specially, on 03:55:00, the knowledge update of *vartrans* in *servicer* federate was triggered at once because object attribute *freight* in *cooperform* of *cooperator* federate was changed (Fig. f). It was not required to be delayed until sending coupling interaction between federates. Similarly, on 02:10:00, *varsup* was updated because *cooperator* federate interacted with non Agent-DEVS model to supply 40 trucks (Fig. j). On 06:30:00, *servicer* federate directly made partial planned amount of supply task according to *varstor* (Fig. c). On 06:40:00, the actual amounts of supply task was changed because *supplier* federate interacted with high priority federate model to supply 60 tents (Fig. g). In the simulation test, not only the dynamic changes of supply process but also the results of time advance represent the facts in a certain extent.



(a)

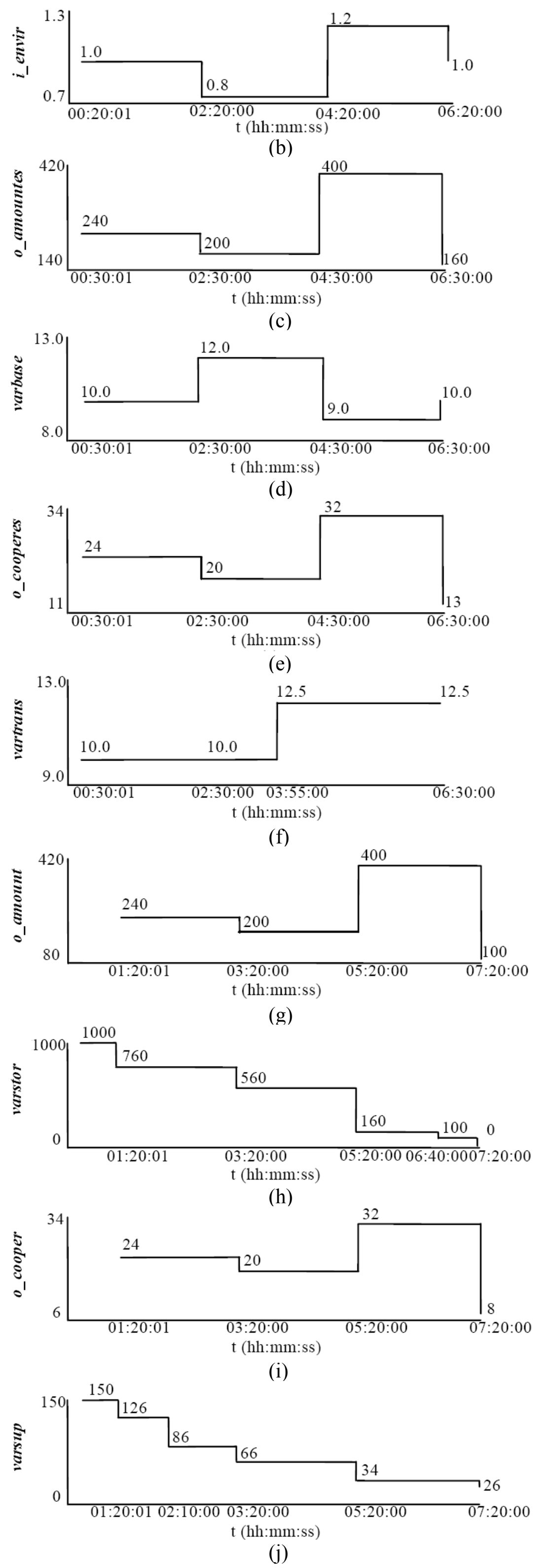


Figure 3. Running status of *servicer* federate

VI. CONCLUSION

The simulation models for material supply in emergent disaster are established based on Agent-DEVS. These models translate all kinds of parameters into variables stored in knowledge repository, which can describe more complex behavior. And the intelligence of models is enhanced. The knowledge may be dynamically modified through mutual cooperation, which improves the model abilities for dealing with transactions independently. The distributed interactive simulation for these models is implemented based on HLA. It can interact with other Agent-DEVS and non Agent-DEVS models. And the reusability of models is enhanced. The knowledge update is separated from coupling interaction of models, which enriches the interoperability of models and improves the update efficiency. The autonomy of models is enhanced significantly. The main problems in modeling and simulation for material supply in emergent disaster are solved in a certain extent, which lays the foundation for the development of simulation training.

ACKNOWLEDGMENT

This work is supported by Natural Science Foundation Project of Chongqing (Project No. CSTC, 2009BB2346). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers.

REFERENCES

[1] V. Balasubramanian, D. Massaguer, S. Mehrotra and N. Venkatasubramanian, “DrillSim: A Simulation Framework for Emergency Response Drills,” Proc. Intelligence and Security Informatics (LNCS), 2006, pp. 237-248.

[2] A. I. Mondlane, “Integrated Risk Response Techniques in Emergency Situations: The Mozambique Floods Case Simulations,” Proc. International Conference on Sustainable Development and Planning, 2005, pp. 1189-1197.

[3] L. Balbis, F. Gaetani, R. Miniciardi, G. Portella, R. Sacile and E. Trasforini, “A Decisional Model for Dynamic Allocation of Resources in Natural Disasters Management,” Proc 3rd International Conference on Computer Simulation in Risk Analysis and Hazard Mitigation, 2002, pp. 243-252.

[4] J. H. Chen, K. P. Zhou, Z. Y. Zhou, X. J. Shu and D. S. Gu, “Crucial Technology on Constructing the Simulation Platform of Mine Safety,” Mining Research and Development, vol. 26, Nov. 2006, pp. 120-125.

[5] J. P. Qiao and Y. B. Chen, “A Rapid Response System to Landslide Disaster,” Journal of Natural Disasters, vol. 13, Jan. 2004, pp. 132-136.

[6] H. F. Guo and Z. W. Li, “Preliminary Discussion of Fire-fighting Simulated Training System,” Fire Science and Technology, vol. 26, Jan. 2007, pp. 86-89.

[7] Q. Cao, Z. S. He and L. Yu, “Research into Simulation and Modeling of Material Supply in Emergent Disaster Based on DEVS/CD++,” Journal of Computer Applications, vol. 28, Nov. 2008, pp. 2967-2969.

[8] Q. Cao, Z. S. He and L. Yu, “Agent-DEVS: An Extended DEVS Formalism for Intelligent Modeling and Simulation,” Proc. the Second International Conference on Modelling and Simulation, 2009, pp. 286-291.

[9] Q. Cao, Z. S. He, Y. W. Huang and H. Y. Bo, “Research on Simulation and Modeling of Camping Service Based on HLA/MAS,” Journal of System Simulation, vol. 21, Mar. 2009, pp. 1474-1478.